Graph Spectral Filtering for Network Simplification

Markus Diego Dias*
Paola Valdivia*
*ICMC - Universidade de São Paulo
São Paulo, Brasil

Fabiano Petronetto†
†Departamento de Matemática
Universidade Federal do Espírito Santo
Espírito Santo, Brasil

L. Gustavo Nonato*‡
‡Tandon School of Engineering
New York University
New York, USA

***Abstract*—Visualization is an important tool in the analysis and understanding of networks and their content. However, visualization tools face major challenges when dealing with large networks, mainly due to visual clutter. In this context, network simplification has been a main alternative to handle massive networks, reducing complexity while preserving relevant patterns of the network structure and content.**

In this paper we propose a methodology that rely on Graph Signal Processing theory to filter multivariate data associated to network nodes, assisting and enhancing network simplification and visualization tasks. The simplification process takes into account both topological and multivariate data associated to network nodes to create a hierarchical representation of the network. The effectiveness of the proposed methodology is assessed through a comprehensive set of quantitative evaluation and comparisons, which gauge the impact of the proposed filtering process in the simplification and visualization tasks.

I. Introduction

Networks (or graphs) are important structures for modeling systems whose elements bear a pairwise relationship.

Among the analytical tools designed to analyze networks, visualization plays a crucial role, assisting users in the understanding of important information such as network communities and their relationship. However, visualization tools face challenges when dealing with large networks. For instance, when large networks are drawn as standard node-link diagrams, the visualization can easily become cluttered, hampering the visual analytic tasks. Network simplification mechanisms have been a main alternative to get around visual clutter, making large networks manageable in terms of visualization. Most network simplification tools rely on node adjacency information to build a hierarchical representation of the network while preserving important structures in each level of the hierarchy. Nonetheless, networks typically carry attributes associated to their nodes, which should not be disregarded during the simplification process. In fact, network attributes can be useful to identify clusters of similar elements and also to investigate the behavior of particular individuals in a global network scenario.

Since most network simplification methods does not account for attribute information, they are prone to group nodes (or edges) with very different content during the simplification process. Moreover, the few techniques able to handle attribute information do not perform well when facing elements whose content are very distinct from other attributes (outliers), resulting simplified networks with a few meta-nodes representing most of the original nodes and several of the "outliers" as single nodes. Therefore, those methods tend to emphasize outliers rather than communities and their relations.

In this work we propose a methodology that improves network simplification tasks while avoiding the issues pointed above. The proposed methodology relies on both topological and attribute information to accomplish the network simplification. More specifically, our approach makes use of spectral filtering schemes derived from Graph Signal Processing (GSP) theory [1]. The proposed filtering process smooths out attributes with similar content and discriminates attributes with distinct information, making the simplification process less sensitive to outliers while better grouping "similar" nodes. Differently from other hierarchical simplification techniques that focus only on the simplification, our methodology also improves the visualization of the attributes, facilitating their understanding.

The effectiveness of our methodology is assessed through a number of experiments and comparisons against other simplification schemes.

In summary, the main contributions of this work are:

- A generalization of Graph Spectral Filtering theory for multivariate data.
- A hierarchical network simplification methodology based on the proposed filtering scheme.
- A study of the effect of the Graph Spectral Filtering on the network simplification process.

II. Related Work

In this section we review existing techniques that rely on node collapse schemes to perform network simplification.

Several existing techniques rely only on network topology information to accomplish the simplification. Good examples of topology-based simplification are Phrase Nets [2] and Compressed Adjacency Matrix [3], which collapse nodes with the same neighbors. Power graph analysis [4]–[6] also collapse nodes that share similar neighbors, but using more relaxed constraints in terms of how to compare the neighborhoods. Other examples of techniques that rely only on topological information to perform network simplification are Ask-Graph View [7], which relies on hierarchical clustering, and Bastian et al [8], which uses persistent homology to detect clique communities on complex networks.

Matching in a graph [9] is another mechanism commonly employed to perform simplification from topological information. Matching is an important tool to create hierarchical

representations so as to leverage graph analysis [10] and visualization [11], [12].

Techniques that rely only on topological information are quite limited as to the type of network they can play with, fostering the development of simplification methods able to handle both topology and network attribute content. Some of those content-based methods make use of optimization schemes to group similar nodes [13]–[16], being spectral clustering an important alternative in this context [17].

Matrix factorization has also been employed to simplify large networks [18]. For instance, the Vegas [19] system uses SymNMF [20] to summarize citation networks. Multivis [21] is an information visualization technique that builds upon tensor decomposition to simplify networks built from email content. Dias et al. [22] rely on Non-Negative Matrix Factorization (NMF) to generate hierarchical representation of networks, comparing the effect of different factorization schemes in simplification process.

In the context of network visualization, several methods have been proposed to visual simplify networks based on their topology and attribute content. OnionGraph [23] and Pivot Graph [24] for instance perform semantic aggregation while Elmqvist et al. [25] propose a visualization scheme that generates a hierarchical representation from node's attributes. However, most hierarchy-based visualization techniques assume that the hierarchy is given as input, accomplishing only the visualization [26], [27].

Techniques described above do not rely on filtering schemes to help the simplification process, being this a major contribution of the present work. Our approach combines graph spectral filtering and matching in a single context, producing hierarchical representations of networks where similar nodes are properly grouped in each level of the hierarchy. Moreover, the proposed spectral graph filtering scheme reduces the noise on the data associated to the nodes of a network, making the visualization of network structures and the node content itself cleaner and easier to interpret.

III. Graph Fourier Transform

Graph Signal Processing (GSP) [1] aims to develop tools for processing data defined on irregular domains such as graphs. The GSP framework has already been used to assist information visualization applications such as the visual analysis of urban mobility data [28] and dynamic networks [29]. Before describing the proposed filtering scheme that will be used to support network simplification tasks, we present the mathematical and computational foundations of Graph Fourier Transform and Spectral Filtering, which are the basis of our methodology.

A. Graph Fourier Transform

Let $G = (V, E)$ be a undirected graph, where V is the set of nodes $\{v_1, v_2, ..., v_n\}$ and E the set of edges $\{(v_i, v_j), i \neq j\}$. A weighted adjacency matrix $A = (a_{ij})$ is a matrix where each entry a_{ij} represents the weight of the edge (v_i, v_j) in E ($a_{ij} = 0$ if the edge is not in E).

A signal is a function $f : V \to \mathbb{R}$ defined on the nodes of G that associates a scalar $f(v_i)$ to each node $v_i \in V$. The signal can be represented as a vector in $\mathbb{R}^n$, where the i_{th} component of the vector represents the signal value at the node v_i.

The (non-normalized) graph Laplacian matrix is given by $L = D - A$, where D is a diagonal matrix with entries d_{ii} equal to the sum of the elements in the i-th row of A. Since A is symmetric, real and positive semi-definite, the graph Laplacian is a real, symmetric, and positive semi-definite matrix. Therefore, it has a complete set of orthonormal eigenvectors $\{u_l\}_{l=1,2,...,n}$ with corresponding non-negative real eigenvalues $\{\lambda_l\}_{l=1,2,...,n}$. Zero appears as an eigenvalue with multiplicity equal to the number of connected components of the graph [30]. Considering a connected graph, the eigenvalues can be ordered as $0 = \lambda_1 < \lambda_2 \leq \lambda_3 \leq ... \leq \lambda_n$ and zero is an eigenvalue whose corresponding eigenvector is a constant vector. The set of eigenvalues $\sigma = \{\lambda_1, \lambda_2, \lambda_3, ..., \lambda_n\}$ is called the spectral domain of L.

Let $u_l(v_j)$ be the the j_{th} coordinate of eigenvector u_l. The Graph Fourier Transform (GFT), $\hat{f} : \sigma \to \mathbb{R}$, of a signal f is defined as the expansion of f in terms of the eigenvectors of the graph Laplacian:

$$\hat{f}(\lambda_l) = \langle u_l, f \rangle = \sum_{j=1}^{n} u_l(v_j) f(v_j) \tag{1}$$

Given the Graph Fourier Transform $\hat{f}$, the signal f can be recovered via the inverse Graph Fourier Transform (iGFT), which is defined as:

$$f(v_i) = \sum_{l=1}^{n} \hat{f}(\lambda_l) u_l(v_i) \tag{2}$$

If we denote by U the (orthogonal) matrix with columns given by the eigenvectors u_l, the GFT and iGFT can be obtained by matrix multiplication as follows:

$$\hat{f} = U^T f \qquad f = U\hat{f} \tag{3}$$

The eigenvalues and eigenvectors of the graph Laplacian play a similar role as frequencies and basis functions in the classical Fourier theory. As said before, zero is an eigenvalue whose corresponding eigenvector is a constant vector (assuming the graph is connected). The eigenvectors associated with low eigenvalues λ_l (low frequencies) vary slowly across the graph; i.e., if two nodes are connected by an edge with a large weight, the values of the eigenvector at those locations are prone to be similar. The eigenvectors associated with larger eigenvalues (larger frequencies) oscillate more and they are more likely to have dissimilar values on adjacent nodes.

B. Graph Spectral Filter

A graph spectral filter is a function $\hat{h} : \sigma \to \mathbb{R}$ that associates a scalar value $\hat{h}(\lambda_l)$ to each eigenvalue $\lambda_l \in \sigma$. We can define frequency filtering, or graph spectral filtering, of a signal f as:

$$\tilde{\hat{f}}(\lambda_l) = \hat{f}(\lambda_l)\hat{h}(\lambda_l), \tag{4}$$

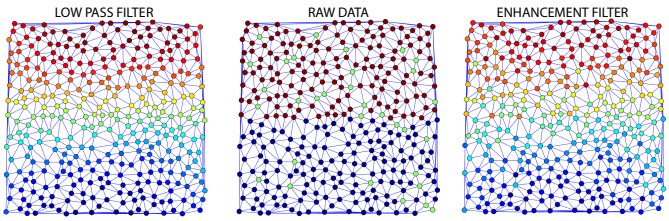

Fig. 1. Central figure represents a graph with signal defined by a step function (the nodes signal are 0 or 1) with noise added (their signal are 0.5). Left and right figures show the result of applying a low pass and an enhancement filter to the step function.

where $\hat{f}$ is the graph Fourier transform of f and $\hat{h}$ is a graph spectral filter. Using the inverse graph Fourier transform we can obtain the filtered version $\tilde{f}$ of f in the graph domain:

$$\tilde{f}(v_i) = \sum_{l=1}^{n} \hat{f}(\lambda_l)\hat{h}(\lambda_l)u_l(v_i). \tag{5}$$

Using the matrix notation, $\tilde{f}$ can be obtained as follows:

$$\tilde{f} = UHU^T f \tag{6}$$

where H is a diagonal matrix with entries $\hat{h}(\lambda_1), ..., \hat{h}(\lambda_n)$.

In this work we will study the effect of two different filters, the low-pass filter and the enhancement filter. We call a filter as low-pass if it does not significantly affect frequencies in the left most part of the spectral domain (low-frequencies) but it attenuates, i.e., reduces, the magnitude of frequencies in the mid-right region of the spectral domain. In contrast, high-pass filters attenuate low-frequencies and preserve high-frequencies. An enhancement filter emphasizes/preserves both the low- and high-frequencies simultaneously, reducing the magnitude of frequencies in the mid part of the spectral domain.

Let λ_{cut} be a real number where $0 < \lambda_{cut} < \lambda_n$. A low-pass filter $\hat{h}_l$ can be defined as:

$$\hat{h}_l(\lambda_l) = \begin{cases} 1, & \text{if} \quad \lambda_l \leqslant \lambda_{cut}, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

For example, the local mean on the adjacent nodes of the graph signal is a low-pass filter. But using different values for λ_{cut} we have different low-pass filters.

Let $0 < \lambda_{cut_1} < \lambda_{cut_2} < \lambda_n$. An enhancement filter $\hat{h}_e$ can be defined as:

$$\hat{h}_e(\lambda_l) = \begin{cases} 1, & \text{if} \quad \lambda_l \leqslant \lambda_{cut_1} \text{ or } \lambda_l \geqslant \lambda_{cut_2} \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

We will use the low-pass filter and the enhancement filter defining as before. The effect of the two filters on a step function with noise can be seen in Figure 1.

IV. Spectral Filtering in Multidimensional Data and Network Simplification

Given a network G, the proposed methodology to build a hierarchical representation of G relies on build a hierarchy by collapsing edges connecting similar (meta)nodes. Each collapse leads to a new meta node representing a pair of collapsed (meta)nodes.

Some issues must be handled when building a hierarchical representation via edge collapse. In order to collapse similar nodes we need first to measure the similarity between adjacent nodes. Our proposal is to use spectral filtering to process attributes associated to the nodes of a network and then calculate the similarity between nodes from the filtered data. However, the spectral filtering theory demands a single scalar value associated to each node of the network, while in our context we may have a vector of attributes associated to each node. Therefore, we need to adapt the spectral filtering theory to this context.

A. Spectral Filtering in Multidimensional Data

In this subsection we show how to adapt the spectral graph filtering to multivariate data.

Let $G = (V, E, F)$ be a network, where V is the set of nodes with $|V| = n$ and E the set of edges. F is an attribute matrix of size $n \times m$ where the i-th row corresponds to the numerical attributes associated with the node v_i. In other words, each entry f_{ij} in F corresponds to the value of the attribute j in the node v_i. Each column of F represents one of the m attributes, so we have an n-dimensional array containing real values for each attribute. We denote by f^j this n-dimensional array containing the j-th attribute (coordinate) associated to the nodes in V. Recalling that a signal on the graph can be represented by an array in $\mathbb{R}^n$, $f^j : V \to \mathbb{R}$ can be seen as a signal defined on G. This signal will be represented by the array f^j and the value of the signal on the node v_i is defined by $f^j(v_i) = f_{ij}$.

Now we can use all the Graph Signal Theory defined above to process each signal f^j by simply replacing the signal f by f^j in the definitions from Section III.

We represent by $\hat{F}$ the matrix $n \times m$ where each column $\hat{f}^j$ corresponds to the GFT applied to the j-th column of F. The value $\hat{f}^j$ in the eigenvalue λ_l is $\hat{F}_{jl} = \hat{f}^j(\lambda_l)$. Therefore, $\hat{F}$ can be calculated by matrix multiplication as follows:

$$\hat{F} = U^T F \tag{9}$$

Given a spectral filter $\hat{h}$, we can compute the filtered version $\tilde{\hat{f}}^j$ of $\hat{f}^j$ in the graph domain via iGFT, (denoted as $\tilde{f}^j$), resulting in the filtered matrix $\tilde{F}$. We will always use the same filter $\hat{h}$ in every signal f^j. Therefore, the matrix $\tilde{F}$ containing filtered signals $\tilde{f}^j$ can be obtained as follows:

$$\tilde{F} = UHU^T F \tag{10}$$

B. Node Similarity.

Given filtered attributes, we have to find similar nodes which will be the candidates to be collapsed during the construction

of the network hierarchical representation. The main issue is how to measure the similarity between adjacent nodes?

Given two nodes v_k and v_s, and their corresponding attributes $\tilde{f}_k$ and $\tilde{f}_s$ (rows in $\tilde{F}$), we can calculate the similarity between those nodes using the Euclidean or Mahalanobis distances between v_k and v_s. Given an edge (v_k, v_s), the weight of this edge will be defined as the value of the similarity between the nodes v_k and v_s. Therefore, collapsing edges with large weights corresponds to merging highly similar nodes.

C. Hierarchical Network Simplification

The hierarchy representing the original network in different levels of detail is built using a methodology similar to the one proposed by Dias et al. [22], which relies on a matching in a graph to collapse nodes (or edges).

Matching in a graph. There are several advantages in collapsing nodes based on matching. The matching tends to point a large number of edges to be collapsed in each step of the hierarchy construction. Moreover, the collapse of an edge does not conflict with the others, making possible to collapse many edges simultaneously.

A subset of edges $M \subset E$ is a matching in G if no two edges in M are adjacent, that is, edges in M do not share a common node. This property guarantees that edges in M can be collapsed without conflicts. A matching M is called maximal if there is no other matching M' such that $M \subset M'$. Given a weighted set of edges E, where $w(e)$ is the weight associated to the edge $e \in E$, let $C(M) = \sum_{e \in M} w(e)$ be the total cost of a matching M, and $\mathcal{M}$ be the collection of all matchings on G. A matching $M \in \mathcal{M}$ is a maximum weighted matching (MWM) if $C(M') \leq C(M)$ for every $M' \in \mathcal{M}$.

Instead of the maximum weighted matching (MWM), we calculate an approximation of MWM, the *sorted maximal matching* (SMM). The SMM is computed by sorting the edges in E in descending order of weights; then a matching set M is built by adding edges to M in the sorted order. If an edge to be added is incident to an edge already in M then it is discarded and the next edge in the sorted list is considered. The process follows until all edges are considered.

The SMM is not guaranteed to be maximal nor of maximum weight. However, it always includes the edge with the largest weight in the matching list, ensuring thus that the two most similar nodes will always be collapsed in each step of the hierarchy construction. Moreover, the computation of SMM is computationally more efficient than the MWM.

Hierarchy. We adopt superscript indexes to represent levels of the hierarchy, $(v_j)_t$ corresponds to a (meta)node in the t-th level of the hierarchy, $t = 0$ is the original network. We denote by $|(v_j)_t|$ the number of nodes from the original graph merged into $(v_j)_t$. Since the row $\tilde{f}_j$ of $\tilde{F}$ corresponds to the row f_j in the attribute matrix F, we define a new filtered matrix $(\tilde{F})_t$ with rows given by:

$$(\tilde{f}_j)_t = \frac{1}{|(v_j)_t|} \sum_{s \in (v_j)_t} \tilde{f}_s \tag{11}$$

In other words, $(\tilde{f}_j)_t$ is the average of the rows in $\tilde{F}$ corresponding to nodes in $(v_j)_t$. Entries in $(\tilde{f}_j)_t$ can also be interpreted as values of a signal on a meta-node. This merging mechanism avoids repeated computation of the signals, GFT, filtering and iGFT in each level of the hierarchy. Therefore, the hierarchy construction is computationally viable and mathematically sound.

V. Datasets and Metrics

The effect of our methodology is assessed using different high-dimensional datasets. Those datasets, jointly with three quality metrics, are used to evaluate and compare the effect of the graph spectral filtering in the simplification process.

A. Datasets

Table I list datasets used in our work with number of nodes and attributes of each dataset. The hierarchical simplification is performed until the finer level contains exactly the number of metanodes in last column. **College Football** dataset [31] provides information about the game table of a College Football Division in 2000. We associate multivariate data to each node i by creating a feature vector x_i with dimension 115, where each entry x_{ij} stores the number of times that the team i played against the team j. The network is constructed by creating edges between nodes (teams) that face each other in the season. **Ecoli** dataset [32] contains protein localization sites. **Iris** dataset [32] is a well known database found in the pattern recognition literature. Each instance is a different iris plant and the attributes represent length/width of the sepal/petal of this iris plant. **Wine** dataset [32] was created using the results of a chemical analysis of wines grown in the same region in Italy. The analysis determined the quantities of 13 constituents found in each of the three types of wines. Another dataset of wines is **Wine Quality** dataset [33]. These data is made of white variants of the Portuguese "Vinho Verde" wine. The numerical attributes represent the results of physicochemical tests for each wine. The network in last four datasets was built using the KNN-graph derived from the dataset using 18, 12, 13 and 69 neighbors respectively. **VIS Conference** dataset [34] contains information of papers published at the IEEE VAST, InfoVis, and SciVis conferences. A node represents an author. Nodes are connected if the authors collaborated at least once. The node information is derived from the titles of the papers authored, as a term-frequency matrix (*bag-of-words*).

	nodes	attributes	clusters
College Football	115	115	12
Ecoli	336	7	8
Iris	150	4	4
Wine	178	13	5
Wine Quality	4898	11	10
Vis Conference	966	458	10

TABLE I

B. Metrics of Validation

The effectiveness of our methodology is assessed using three different metrics from other works, modularity, Δ–Measure, and K-Way Ratio Cut Cost Metric. These three metrics quantify the quality of clusters on a graph. In our tests, each meta-node in a coarser level of the hierarchy is considered as a cluster comprising nodes from the original network, allowing the use of those metrics.

Modularity. It was used by Newman [13] and Wang et al. [18] to validate their simplification methods. Networks with high modularity have dense connections within defined clusters and sparse connections among different clusters. Assuming that the nodes are labeled according their cluster, let e_{ij} be the fraction of edges connecting nodes from cluster i to cluster j and $a_i = \sum_j e_{ij}$. A partition $\Phi = \{\mathcal{G}_1, \mathcal{G}_2, ..., \mathcal{G}_k\}$ of a network is a set of smaller components that divides the network. The modularity Q of a partition Φ is:

$$Q(\Phi) = \sum_i (e_{ii} - a_i^2) \tag{12}$$

where $Q = 0$ indicates random groupings and $Q = 1$ indicates the maximum modularity, created by well structured clusters.

Δ–Measure. The Δ-measure [14], [15] assesses the quality of group formation by measuring pairwise relationships between the clusters. Let $\Phi = \{\mathcal{G}_1, \mathcal{G}_2, ..., \mathcal{G}_k\}$ be a partition of the nodes from G such that $\mathcal{G}_i \cap \mathcal{G}_j = \emptyset$, for all $i \neq j$, and

$$P_{\mathcal{G}_j}(\mathcal{G}_i) = \{u |\ u \in \mathcal{G}_i \text{ and } \exists\ v \in \mathcal{G}_j \text{ s.t. } (u, v) \in E\}.$$

Making $p_{i,j} = (|P_{\mathcal{G}_j}(\mathcal{G}_i)| + |P_{\mathcal{G}_i}(\mathcal{G}_j)|)/(|\mathcal{G}_i| + |\mathcal{G}_j|)$ we define the Δ-measure as:

$$\Delta(\Phi) = \sum_{\mathcal{G}_i, \mathcal{G}_j \in \Phi} = (\delta_{\mathcal{G}_j}(\mathcal{G}_i) + \delta_{\mathcal{G}_i}(\mathcal{G}_j)) \tag{13}$$

where $\delta_{\mathcal{G}_j}(\mathcal{G}_i) = |P_{\mathcal{G}_j}(\mathcal{G}_i)|$ if $p_{i,j} \leq 0.5$ and $\delta_{\mathcal{G}_j}(\mathcal{G}_i) = |\mathcal{G}_i| - |P_{\mathcal{G}_j}(\mathcal{G}_i)|$ otherwise. To obtain the average contribution of the clusters, we divide it by k. The smaller the result, the better the cluster formation.

K-Way Ratio Cut Cost Metric. The K-Way Ratio Cut Cost Metric [17] measures the cost of a graph cut generating a k-way partition $\Phi = \{\mathcal{G}_1, \mathcal{G}_2, ..., \mathcal{G}_k\}$. Let E_h be the sum of the weights of the edges with exactly one end in $\mathcal{G}_h$. The cut cost can then be defined as:

$$cost(\Phi) = \sum_{h=1}^{k} \frac{E_h}{|\mathcal{G}_h|} \tag{14}$$

The smaller the K-Way Ratio Cut Cost, the better the partition.

VI. Results and Comparisons

In this section we present a comprehensive set of experiments and comparisons that assess the impact of the proposed filtering methodology into the network simplification process. Specifically, our experiments aim to answer the following questions:

1) How does spectral filtering impact into the network simplification process?

2) How is the network topology affected in each level of the hierarchy?

3) How does the filtering process impact the visualization of attributes?

All datasets used for comparison in this paper are small for network simplification, which usually involves massive networks. Our objective in these comparisons was to show the effect of filtering the data. The matching set is defined to choose edges on all the network, not only on one region of the network. This global behavior allows to simplify all the network homogeneously independent of the size of the network. As previously mentioned, in our tests and examples we fix the number of nodes in the coarser level of the hierarchy.

A. Node-Link Visualization and Clustering Behavior

In the following we compare node-link visualizations of the network simplification process involving filtered and raw data. Nodes are colored according to the meta-node in the coarser level of the hierarchy. More specifically, nodes belonging to the same meta-node on the coarser level have the same color in all levels of the hierarchy. The size of each meta-node reflects the number of nodes collapsed on it.

Figure 2 depicts four levels of the hierarchical network simplification process for the Ecoli dataset using both raw (top row) and low-pass filtered (bottom row) data. Notice that the size of the meta-nodes in the coarser level of the hierarchy differs considerably when using raw and filtered data. The meta-nodes resulting from low-pass filtered data are more homogeneous in terms of their size. In contrast, when using raw data, the simplification process groups most of the original nodes in a few meta-nodes while producing meta-nodes containing only one or two of the original nodes.

A similar behavior can be seen in Figure 3. When raw data is used, meta-nodes with a few nodes are produced in the coarser level of the hierarchy. More homogeneous meta-node sizes are produce when low-pass filtered data is employed, preventing that large meta-nodes show up in the process.

We are not searching necessarily for clusters with similar sizes, but avoid clusters with very few nodes. Because we are trying to find a desired number of clusters. Only two or three nodes can not represent a real cluster. The simplification with the low-pass filter doesn't present the clusters with similar sizes always as can be seen on section VII.

B. Attribute Visualization

In this section we show the impact of the filtering mechanism when visualizing the attributes associated to the nodes. To this end we rely on parallel coordinates as visual metaphor, where each node in the original network corresponds to a polyline in the parallel coordinates diagram. Nodes belonging to the same meta-node in the coarser level have the same color in the diagram.

The Iris Dataset Network has been built and simplified such that only three meta-nodes remains in the coarser level. The corresponding parallel coordinates visualization is depicted in Figure 4. Notice that in this case the simplification using raw and filtered data results in similar visualizations, what

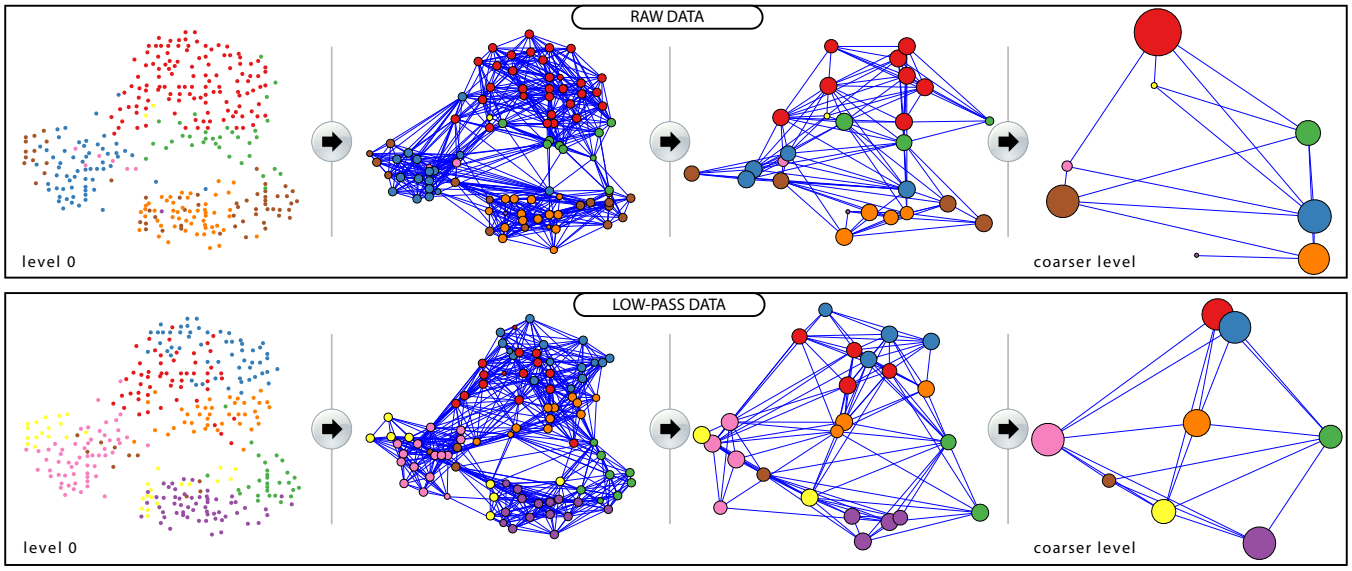

Fig. 2. Network Simplification steps of the ecoli dataset using raw data (top) and low-pass filtered data (bottom).

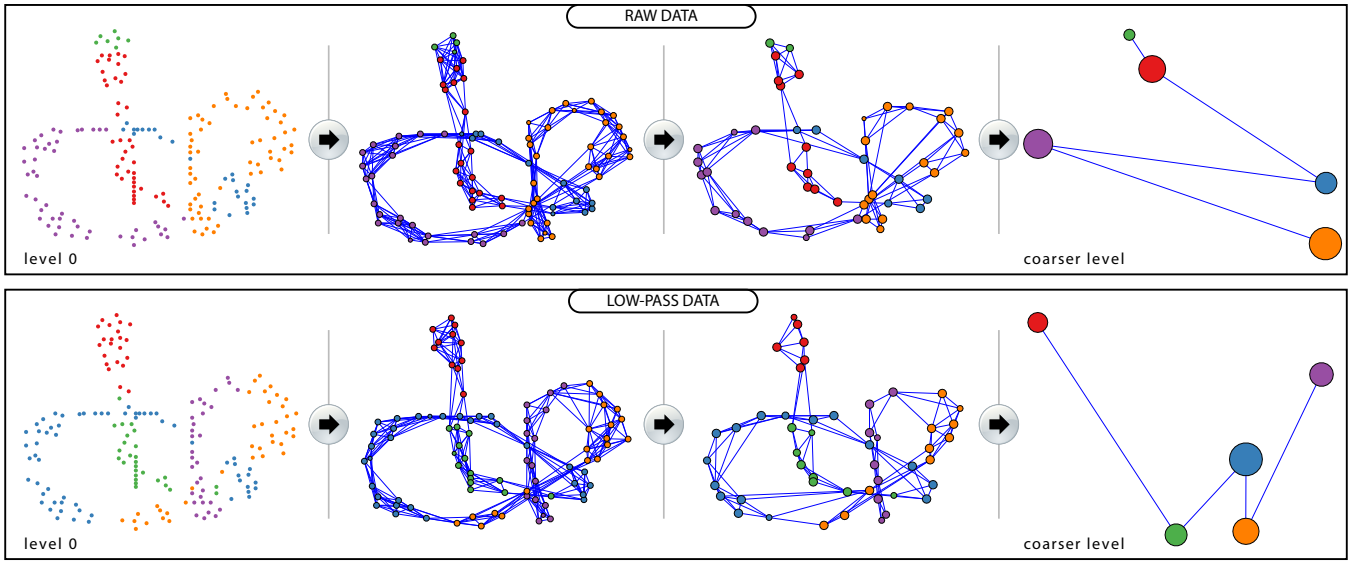

Fig. 3. Network Simplification steps of the wine dataset using raw data (top) and low-pass filtered data (bottom).

is expected, as the attributes in the Iris data set can already discriminate the groups.

Attributes from the Ecoli dataset, in contrast to the Iris dataset, can not clearly discriminate the groups, as one can notice from the parallel coordinates visualization depicted on left most image in Figure 5. The mid and right images in Figure 5 show visualizations from simplifications accomplished using filtered data. Notice that groups are better defined in the visualizations involving filtered data, mainly in the low-pass filtered case, showing that the simplification is grouping together nodes with similar content. The enhancement filter also emphasizes high-frequencies, so preserving differences while bringing closer similar nodes.

Figure 6 shows a similar analysis but using the Wine dataset simplified until three (top) and five (bottom) meta-nodes remains in the coarser lever of the hierarchy. Notice that while the visualization using raw data does not allow to clearly identify the groups, the visualizations resulting from filtered data are much cleaner, revealing the groups. Once again, groups are better defined when low-pass filtered data is used.

C. Quantitative Evaluation

In this section we evaluate the simplification process using quantitative measures. The goal is to gauge whether spectral filter negatively impacts in the quality of the simplified network. As we shall show, that is not the case, that is, the quality of the networks simplified from filtered data are better or similar than the ones simplified using raw data.

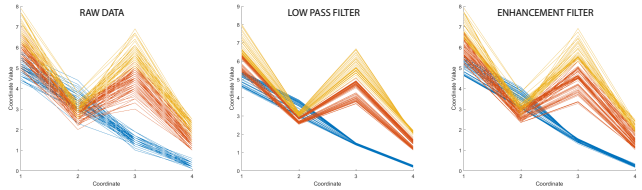

Fig. 4. Iris Dataset simplified until three meta-nodes remain in the coarser level.

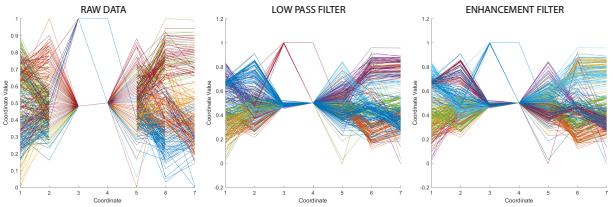

Fig. 5. Ecoli Dataset.

Figure 7 left shows the result of applying the modularity metric in networks simplified from raw and filtered data. Notice that the modularity from filtered data is better for the Ecoli, Iris and Wine datasets. For the Iris dataset the difference was very small. For the Wine Quality and College Football datasets the results are practically the same.

Similar results are observed when using the Δ-measure (Figure 7 center) and the K-Way Ratio Cut Cost Metric (Figure 7 right). In fact, the better quality of simplifications resulting from filtered data is even easier to notice when Δ and K-Way Ratio metrics are used as quality measures.

VII. Discussion and Limitations

The proposed methodology turned out to be quite effective in most of the experiments we performed, showing that our choice of filtering data before accomplishing network simplification is an attractive alternative. We are not saying that is always better to filter the data, but that spectral filters are an alternative to be considered when simplifying networks based on both topology and attribute information.

The design of optimal filters to operate in specific datasets is a problem that we are planning to investigate in a follow up work. In fact, a multitude of filtering schemes can be explore, being spectral graph filtering theory a rich research field.

For networks with a big central cluster and isolated small clusters like the VIS Conference network (Figure 8), the proposed methodology for simplification may not work perfectly, since two small clusters have left on the finer level when we simplified the network until we have ten metanodes. But comparing our result with the HNMF method [22], the HNMF without a filter presented a higher number of isolated small metanodes. Basically dividing the dataset on only four clusters. Using our methodology to filter the attributes allied with the HNMF, we can identify eight clusters well defined.

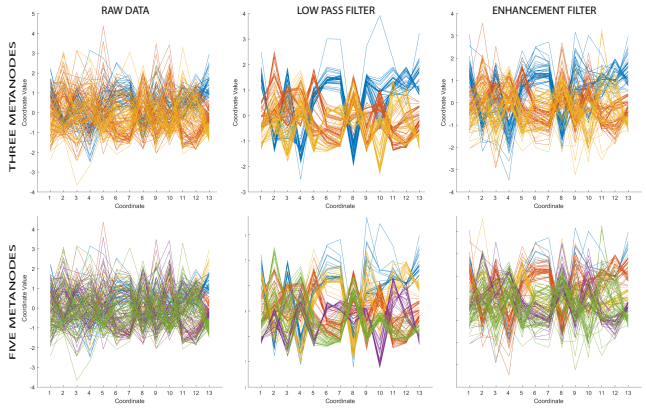

Fig. 6. Wine Dataset simplified until three (top) and five metanodes (bottom) remains in the coarser level.

VIII. Conclusion

We proposed a generalization of the theory of Graph Spectral Filtering to the context of network simplification. We provided a new methodology to perform and improve hierarchical network simplification using attribute data associated to the nodes of the network. Our methodology relies on graph signal processing to filter the data before simplification. The proposed methodology was assessed and compared against simplifications performed using raw data. The results show that simplifications using filtered data tends to be of better quality when compared against the ones involving raw data, showing our approach is an attractive alternative.

use section* for acknowledgment

Acknowledgment

Markus acknowledges CAPES for the financial support.

References

[1] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 16. 83–98, 2013.

[2] F. van Ham, M. Wattenberg, and F. B. Viégas, "Mapping text with phrase nets." *IEEE TVCG*, vol. 15, no. 6, 2009.

[3] K. Dinkla, M. A. Westenberg, and J. van Wijk, "Compressed adjacency matrices: Untangling gene regulatory networks." *IEEE TVCG*, 2012.

[4] V. Yoghourdjian, T. Dwyer, G. Gange, S. Kieffer, K. Klein, and K. Marriott, "High-quality ultra-compact grid layout of grouped networks," *IEEE TVCG*, vol. 22, no. 1, pp. 339–348, 2016.

[5] T. Dwyer, N. H. Riche, K. Marriott, and C. Mears, "Edge compression techniques for visualization of dense directed graphs," *IEEE TVCG*, vol. 19, no. 12, pp. 2596–2605, 2013.

[6] T. Dwyer, C. Mears, K. Morgan, T. Niven, K. Marriott, and M. Wallace, "Improved optimal and approximate power graph compression for clearer visualisation of dense graphs," *2014 IEEE PacificVIS*, 2014.

[7] J. Abello, F. van Ham, and N. Krishnan, "Ask-graphview: A large scale graph visualization system." *IEEE TVCG*, 2006.

[8] B. Rieck, U. Fugacci, J. Lukasczyk, and H. Leitte, "Clique Community Persistence: A Topological Visual Analysis Approach for Complex Networks," *IEEE Transactions on Visualization and Computer Graphics*, 2017.

[9] G. Karpis and V. Kumar, "A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 359–392, 1998.

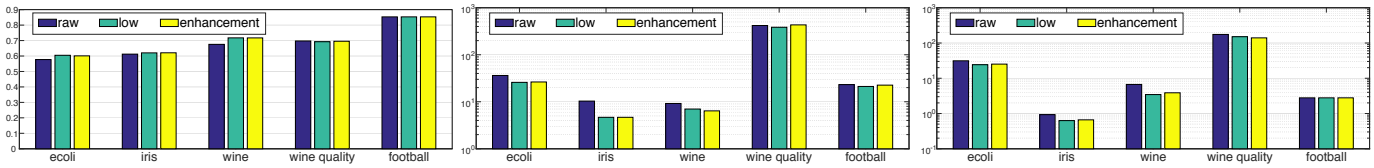

Fig. 7. Comparison of the simplification using the raw data and the filtered data. The blue bar represents the results of the raw data using the modularity metric (left), Δ-Measure (center) and K-Way Ratio Cut Cost Metric (right) for each dataset. The green bar represents the results of the data filtered with the low-pass filter. The yellow bar represents the results of the enhancement filter. The best result on modularity metric is the largest number (less than 1) while on the Δ-Measure and K-Way Ratio Cut Cost Metric, the best results are the smallest.

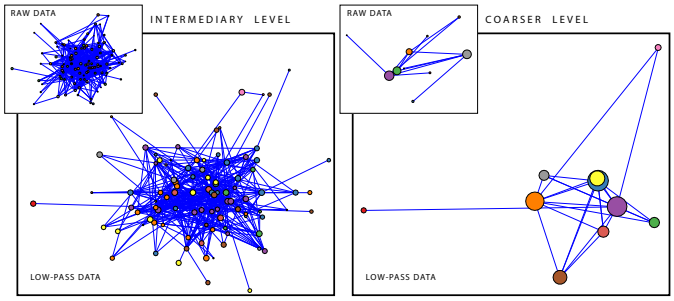

Fig. 8. Comparison of our Graph Spectral Filter methodology allied with the HNMF and the HNMF without a filter using the VIS Conference Network.

[10] E. R. Gansner, Y. Koren, and S. C. North, "Topological fisheye views for visualizing large graphs." *IEEE TVCG*, 2005.

[11] C. Walshaw, "A multilevel algorithm for force-directed graph drawing," in *Proceedings of the 8th International Symposium on Graph Drawing*, ser. GD '00. London, UK, UK: Springer-Verlag, 2001, pp. 171–182.

[12] R. Hadany and D. Harel, *A Multi-Scale Algorithm for Drawing Graphs Nicely*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999.

[13] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Physical Review E*, vol. 69, September 2003.

[14] Y. Tian, R. A. Hankins, and J. M. Patel, "Efficient aggregation for graph summarization." in *SIGMOD*. ACM, 2008, pp. 567–580.

[15] N. Zhang, Y. Tian, and J. M. Patel, "Discovery-driven graph summarization." in *ICDE*. IEEE Computer Society, 2010.

[16] S. Navlakha, R. Rastogi, and N. Shrivastava, "Graph summarization with bounded error." in *SIGMOD*, J. T.-L. Wang, Ed. ACM, 2008.

[17] P. K. Chan, M. D. F. Schlag, and J. Y. Zien, "Spectral k-way ratio-cut partitioning and clustering." *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 13, no. 9, pp. 1088–1096, 1994.

[18] R.-S. Wang, S. Zhang, Y. Wang, X.-S. Zhang, and L. Chen, "Clustering complex networks and biological networks by nonnegative matrix factorization with various similarity measures," *Neurocomputing*, 2008.

[19] L. Shi, H. Tong, J. Tang, and C. Lin, "Vegas: Visual influence graph summarization on citation networks." *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 12, pp. 3417–3431, 2015.

[20] D. Kuang, H. Park, and C. H. Q. Ding, "Symmetric nonnegative matrix factorization for graph clustering." in *SDM*. SIAM / Omnipress, 2012.

[21] J. Sun, S. Papadimitriou, C.-Y. Lin, N. Cao, S. Liu, and W. Qian, "Multivis: Content-based social network exploration through multi-way visual analysis." in *SDM*. SIAM, 2009, pp. 1064–1075.

[22] M. D. Dias, M. R. Mansour, F. Dias, F. Petronetto, C. T. Silva, and L. G. Nonato, "A hierarchical network simplification via non-negative matrix factorization," in *2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, vol. 00, Oct. 2017, pp. 119–126.

[23] Q. Liao, L. Shi, H. Tong, Y. Hu, Y. Zhao, and C. Lin, "Hierarchical focus+context heterogeneous network visualization," *2014 IEEE PacificVIS*, vol. 0, pp. 89–96, 2014.

[24] M. Wattenberg, "Visual exploration of multivariate graphs," in *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, 2006, pp. 811–819.

[25] N. Elmqvist, T.-N. Do, H. Goodell, N. Henry, and J.-D. Fekete, "Zame: Interactive large-scale graph visualization." in *PacificVis*, 2008.

[26] C. Vehlow, F. Beck, and D. Weiskopf, "Visualizing dynamic hierarchies in graph sequences," *IEEE TVCG*, 2016.

[27] R. Blanch, R. Dautriche, and G. Bisson, "Dendrogramix: A hybrid tree-matrix visualization technique to support interactive exploration of dendrograms," in *2015 IEEE PacificVIS*, April 2015, pp. 31–38.

[28] P. Valdivia, F. Dias, F. Petronetto, C. T. Silva, and L. G. Nonato, "Wavelet-based visualization of time-varying data on graphs," in *VAST*. IEEE Computer Society, 2015, pp. 1–8.

[29] A. D. Col, P. Valdivia, F. Petronetto, F. Dias, C. T. Silva, and L. G. Nonato, "Wavelet-based visual analysis for data exploration," *Computing in Science and Engineering*, vol. 19, no. 5, pp. 85–91, 2017.

[30] A. D. Marsden, "Eigenvalues of the laplacian and their relationship to the connectedness," 2013.

[31] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *National Academy of Sciences*, 2002.

[32] D. Dheeru and E. Karra Taniskidou, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[33] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis, "Modeling wine preferences by data mining from physicochemical properties," *Decis. Support Syst.*, vol. 47, no. 4, pp. 547–553, Nov. 2009.

[34] P. Isenberg, F. Heimerl, S. Koch, T. Isenberg, P. Xu, C. Stolper, M. Sedlmair, J. Chen, T. Möller, and J. Stasko, "Visualization publication dataset," 2015.